5 4185

63-4-4

(1)

(6) No title Scale 5 7 + 8 NA

(9) Monthly Progress Report No. 9,

(11) 29 Jul[redacted]63

(12) [12] p.

(10) NA

(13) + (14) NA

TO

Headquarters
Space Systems Division
Air Force Systems Command
Air Force Unit Post Office
Los Angeles 45, California
ATTENTION: SSTK

(16) - (19) NA

(20) U

(21) NA

(15)

In Accordance With
Exhibit "A" to Contract AF04(695)-210
Item VI, Reports

for the Reporting Period

1 [redacted] - 28 Jul[redacted]63,

Project Engineer — I TERRIS

Project Manager — T J BURNS

Contract Administrator — L P GOSLING

Digital Laboratory
Guidance and Controls Division
Aerospace Group
Hughes Aircraft Company
Culver City, California

(4) $1.60

TABLE OF CONTENTS

1.0 Technical Discussion

1.1 Introduction

The effort for July 1963 was in the areas of:

1. Reliability Design for Computer Organizations, and

2. Reliability Programming.

1.2 Reliability Design for Computer Organizations

Work on the sequences required for self-reorganization has been completed and documented as Attachment A to this report. Covered therein, are program sequences after minor failures and master machine sequences after major failures. Also, the physical characteristics of the master and its interconnection with the general-purpose machine are discussed.

A method for increasing the reliability of self-reorganizing electronic digital computers by employing techniques for tolerating common component failures in the basic flip-flop has been developed. This resulted from a study of component reliability and the effect of component failures on circuit operation in light of the logical functions to be performed. It was shown that approximately 80% of all component malfunctions in flip-flop circuits occur in one of the output amplifiers, leaving the other output and the bistable portion in operable condition. Using this fact, gating arrangements and programming techniques were developed for tolerating such faults.

As shown on the milestone chart, work on this task has been completed.

1.3 Reliability Programming

It has been discovered that the Set State Register instruction in the self-reorganizing machine has the effect of creating additional paths for information flow when used properly. That is, when reorganization occurs, the information in the register which is moved in the organization moves with the register. Thus, information has moved thru a new path and allows logic which was classified as essential to proper operation to be classified as nonessential.

The only logic or information paths which can be utilized are those which are under program control and, thus, the concept of microprogramming may be viewed in a new light.

It has been determined that the MTBCF to MTBF ratio of the sample machine is raised to 2.7 from 2.3 when viewed in this new manner. In addition,

it is estimated that microprogram techniques applied to this machine will raise the ratio to 3.3. With optimum application of these ideas, a ratio of 4 is hoped to be possible.

The sacrifice for this advantage is a small amount of memory and speed. The latter is only sacrificed after a logic failure.

A detailed development of a unified error-detecting correcting process is still underway. The general approach previously described has evolved into a more specifically described process as a result of a study of certain fundamental techniques. A combination of these techniques, which will provide an effective error-detecting process, is presently being described.

It was necessary to generate a complete classification of errors upon which to base the detailed techniques. The study has included many and varied items, such as the dependency of the detection method on time and on the functional location of the fault in the machine. The reasonableness check has been studied in some detail along with schemes for testing the memory. Other items include using alternate (or repetitive) computational methods; "offset checking;" short test problems; program check points; checking all possible commands with all possible options; multiplication as a checking tool; tests based on symbolic logic statements; testing the accumulator, adder; hardware checking devices; "address coding;" testing the addressable devices; prevention of excess looping; forced periodic branching; and error counters.

Emphasis has been placed on analyzing the effectiveness of the detecting, correcting process, both during fault-free operation and operation after error detection.

1.4 Follow-On Effort

Work in the above area of reliability programming will continue. Work in the areas of recommended follow-on effort and on the final report will commence.

1.5 Action Items

None.

2.0 Trips and Visits

On 22 July 1963, Mr I Terris of HAC visited Mr J Y Miyamoto of Aerospace Corporation to discuss the technical progress of this study contract.

3.0 Financial, Manpower and Milestone Status

See next page.

3.3 Assigned Personnel

J L Drayer, Member of Technical Staff

C A Finnila, Member of Technical Staff

I Terris, Project Engineer

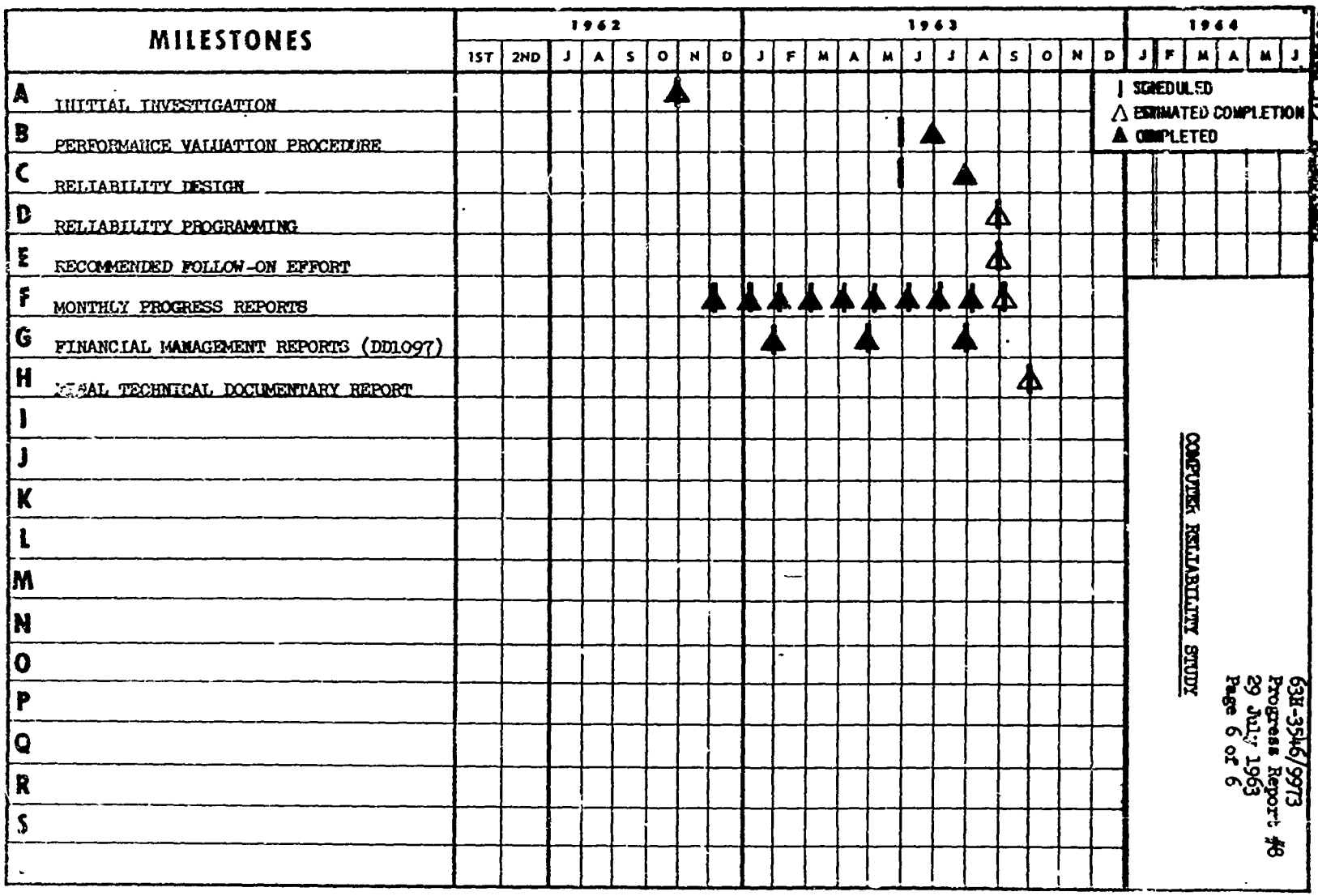
3.4 Additional Information

The dollar and manpower data used in the "Actuals" columns of this report were taken from Hughes Aircraft Company Financial Report Number 980-68 dated 28 July 1963.

3.5 Milestone Chart

The Milestone Chart can be found on the last page of this report.

THIS
PAGE
IS
MISSING
IN
ORIGINAL
DOCUMENT



Section 3.5 MILESTONES

ATTACHMENT A

SELF-REORGANIZING MACHINES - METHODS OF REORGANIZATION

1. Introduction

In a previous report*, the concept of self-reorganizing machines which eliminate unreliable components from the paths of information flow was introduced. The previous report was concerned with the mean time between catastrophic failures (MTBCF) and, by considering the hardware in the main path of the reliability diagram (essential equipment), it was demonstrated for a sample computer that self-reorganization increases the MTBCF by a factor of 2.3.

Previously, only the reorganization logic and equipment were explored and no mention of the sequences or methods of accomplishing self-reorganization, except to say it would be discussed later, was made.

In this report, the methods or sequences required for machine self-reorganization are discussed and applied to the model machine of the previous report.

2. General Philosophy

Malfunctions may be classified in many manners; i.e., duration, effect. For the purposes of this report, classification by effect on operability of the machine is most appropriate. That is, malfunctions shall be divided into two classes:

a. Minor - those which leave the machine almost operating properly, and

b. Major - those which do not.

Minor malfunctions leave the machine in a state where self-reorganization is possible, while major malfunctions leave the machine in a state from which no self-reorganization or diagnosis routines could be run. For recovery from the latter type malfunction, it is necessary to have a master unit which assumes command of the machine, diagnoses the failure, compensates for the malfunction, and returns the machine to the self-operative state. Monitoring of the machine by a master is on a continual or periodic basis, depending on the allowable down-time.

Since most of the master machine is not required for successful operation of the general-purpose machine, the reliability of a majority of the master does not enter into the first order computation of the MTBCF. Only certain failure modes in the circuits of the master enter into the first order approximation of the MTBCF. Failures in a majority of the master may be catastrophic only if a major malfunction also occurs in the general-purpose computer. However, this requires two failures and only appears in the higher order terms of the MTBCF computation.

---

*Attachment A of Monthly Progress Report #6, or IDC 2924.01/10, Techniques for Improving Computer Reliability by Logical Design (Self-Reorganizing Machines)

The over-all program flow chart for a self-reorganizing computer is shown in Figure 1. The programs are grouped as:

a. Tactical,

b. Test,

c. Diagnosis, and

d. Self-reorganizing.

A control program governs entry to the various tactical programs and generates an output to the master. This output is used by the master to detect major malfunctions. The control program determines entry to the various tactical routines and the test routine which must be entered periodically. Each tactical routine may have a reasonableness check to aid in error detection. This check may be used to determine actions of the control and test programs.

The test routine has an output to the master for aid in major malfunction detection and tests by typical self-test routines the integrity of the computer.

When minor malfunctions are detected, the diagnosis routine is used to pinpoint the malfunction, if possible, and determine whether self-reorganization is desired. If self-reorganization is desired, the self-reorganization routine is run.

In the case of a major malfunction, the computer may not provide the proper outputs to the master. In this event, the master performs the reorganization via the state register and starts the computer at the beginning of the test routine as shown in Figure 1.

3. Reorganization Sequences

As indicated above, there are two reorganization sequences, one for minor failures in which the machine still can perform its program, and a second for major failures in which the machine no longer can function properly. Generally speaking, malfunctions in the arithmetic unit, input-output unit, and parts of the control unit, are minor malfunctions and leave the machine in an operational state. Malfunctions in the memory, most of the control unit, and power supply are generally of the major category and render the machine inoperative. In this latter case, the master unit assumes command and reorganizes the machine so that it can operate again. The master machine detects a major malfunction by watching outputs of the machine which must be varied in a predetermined manner by a positive action of the machine to prevent take-over by the master machine. In the event of a major machine malfunction, there is a high probability that the outputs are not properly presented to the master.

3.1 Reorganization Sequences after a Minor Malfunction:

A minor malfunction may be detected by built-in hardware checks such as a parity check or instruction tagbits to insure program sequence, or by a programmed check such as a reasonableness check or an alternate computation. After a minor malfunction is detected, a diagnosis routine determines whether self-reorganization is required and where. Examining the sample machine in the referenced report, it is determined that malfunctions in the arithmetic unit, input-output unit, and the index register, shift counter, and spare counter of the control unit would be considered minor malfunctions.

In this report, malfunction detection and classification shall not be discussed. The self-reorganization program causes self-reorganization by setting the proper state flip-flops as discussed in the previous report.

The self-reorganization routine is entered from the diagnosis routine which has localized the malfunction and determined that self-reorganization is required. To understand the construction of the self-reorganization routine, it should be realized that this routine must be run, at least in part, on a malfunctioning machine. Thus, the routine is constructed so that it is set up prior to malfunction occurrence and can perform any anticipated reorganization without use of those portions of the machine in which a minor malfunction can occur, as shown in Figure 2. After reorganization, the routine would be set up for reorganization after a second failure.

The above is accomplished by the use of a reorganization routine which for the first portion has only SSR (Set-State Register) and UCJ (Unconditional Jump) instructions. These do <u>not</u> use any equipment in which a permanent minor malfunction may occur. The diagnostic enters at the proper SSR instruction and execution of one or several of these instructions performs the needed self-reorganization. Each SSR set is followed by a UCJ to the second half of the routine which modifies the SSR instructions of the first part for a later possible reorganization. The modification of the SSR instructions, which require use of the arithmetic unit, are required for choice of an alternate register in the sample computer.

As an example, let us consider the loss of the index register (XR). The first alternate would be the least significant half of the Q register and the second alternate the least significant half of the A register. At the start, the self-reorganization routine for index register replacement would be:

1.	SSR	22	1	-	Set State Register	22	to	1
2.	SSR	2	0	-	Set State Register	2	to	0
3.	SSR	1	0	-	Set State Register	1	to	0
4.	UCJ	--------						

The first action causes the splitting of the arithmetic unit and sets up most significant half arithmetic. The next two SSR instructions pick Q LSH fo:

the XR replacement. Now, we modify instruction 2 in the second half of the routine to be

SSR 2 1

so that if Q LSH fails, repetition of this same portion of the routine (replacement of XR) will result in A LSH being used for the XR. Note that the diagnostic enters the self-reorganization routine at the same place on a second failure of the XR function, and thus, itself, does not require any modification.

3.2 Sequences for Reorganization after Major Malfunction:

As defined above, a major malfunction would be such that the computer would fail to operate in a nearly normal manner. That is, the computer would be incapable of running a self-diagnosis and causing reorganization to permit correct computational operation. In this case, the positive action required to prevent the master unit from taking over operation would no longer be performed and the master unit would come in to perform the necessary detection, diagnosis, and reorganization.

4. Master Unit

To gain a feeling for the size and complexity of the master unit, a look at the functions which it performs, the manner in which they are performed, and the required implementation is taken.

Referring to the simple computer in the reference, it is seen that the components which can cause major malfunction are the bit and word counter, the program counter, the instruction register, the memory address register, the memory, the memory register, the logic, and the clock pulse generator. It is obvious that some malfunctions would cause the computer to lose all computation capability, others result in a partial but major loss, and others result in a complete stoppage of the machine.

As previously mentioned, the master monitors the computer by watching the setting of certain outputs as shown in Figure 1. If these outputs are not manipulated in a prescribed manner, the master assumes a major malfunction has occurred.

4.1 Loss of Clock Pulses

Although no mention was made of the clock pulse supply for the sample computer, it is reasonable to assume that if an alternate supply were available, one would monitor the clock pulses. (These clock pulses are fed to the various registers, each of which have a gated clock pulse amplifier and loss of the clock pulses completely stops the machine.)

Detection of clock pulse generator failure may be by monostable multivibrators, one for each generator, which would be allowed to reset when clock pulses have been missing for some predetermined time. The resetting of a multivibrator would cause the other generator to be used as the clock pulse source.

4.2 Loss of Program Sequence

Loss of program sequence may be caused by either a permanent or nonpermanent malfunction, and may or may not result in a major malfunction. One could envision a nonpermanent malfunction causing data to be interpreted as an order and the computer to enter a loop of meaningless calculation, thus generating a major malfunction. On the other hand, a permanent malfunction may effect only certain instructions and result in a loss of program sequence which is minor.

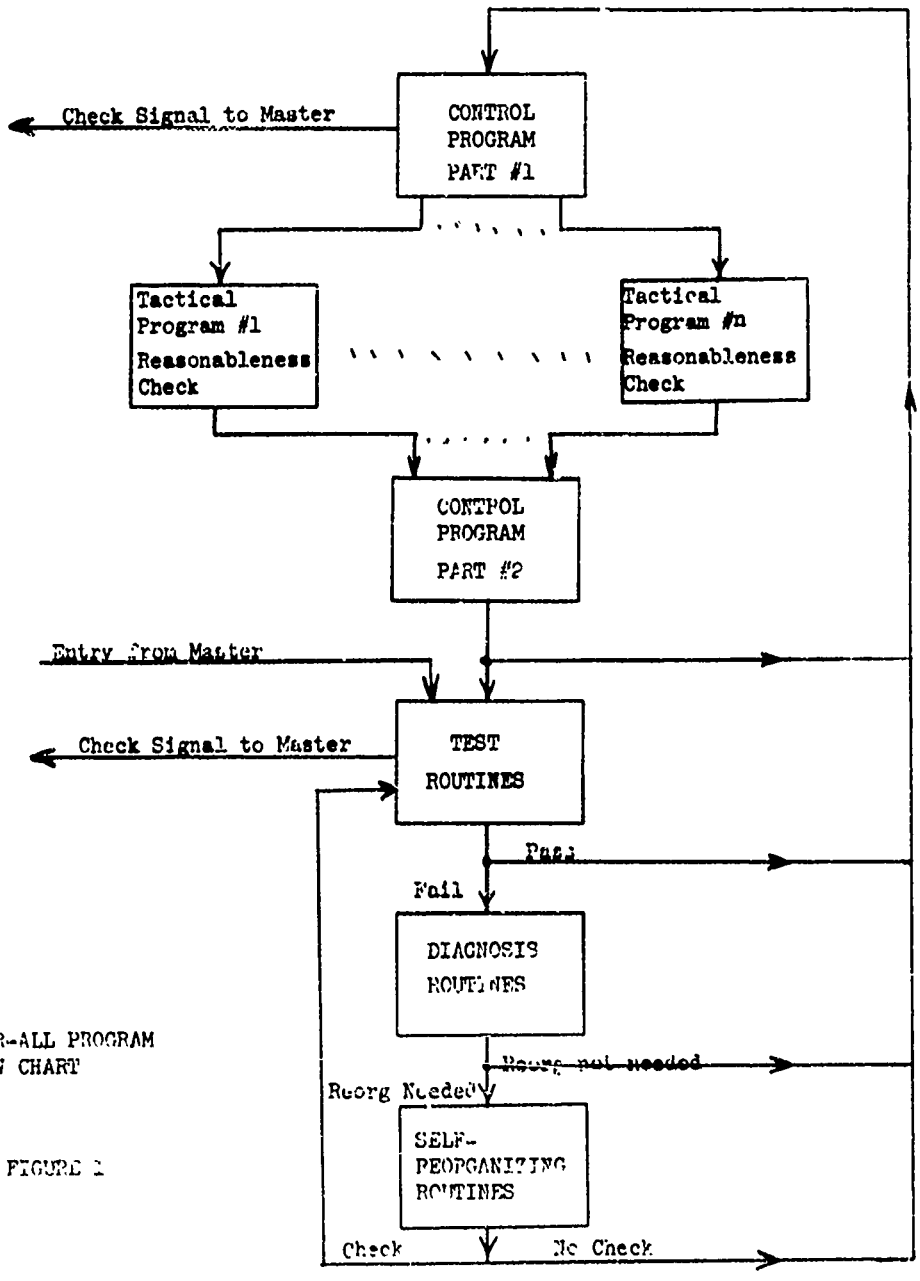
Program sequence may be monitored externally by the use of outputs which must be varied in a prescribed manner by the program. How many outputs and the complexity of their variation depends on the machine and required probability of malfunction detection required. The latter requirement is mission and time-dependent.

A simple scheme would use one output with a monostable multivibrator which is hit every time the test routine is run and a second output which is hit with a prescribed minimum frequency by the other programs. This scheme may be enlarged by use of more outputs. The two output scheme guarantees at least periodic entry to the test routine and periodic entry to a tactical routine. The test routine should be complete enough to guarantee correct operation if entry is achieved and negate the need for more outputs.

If program sequence is lost, the master machine starts the computer at the beginning of its test routine. If the malfunction was nonpermanent, operation will continue in the normal manner. If the malfunction was permanent and the machine is capable of operation (minor malfunction), the test routine will reorganize the machine. If the malfunction is permanent and major, program sequence will be lost again and the master will perform the necessary reorganization. The manner in which this reorganization is performed depends upon the complexity of the master. If the master has no diagnostic capability, reorganization can be by trial and error. Since there are only eight registers which can cause a major malfunction, repair could be achieved very quickly on a trial and error basis if only one register fails at a time. After each reorganization trial the PC is set to the address of the first order in the test routine and the machine is set free.

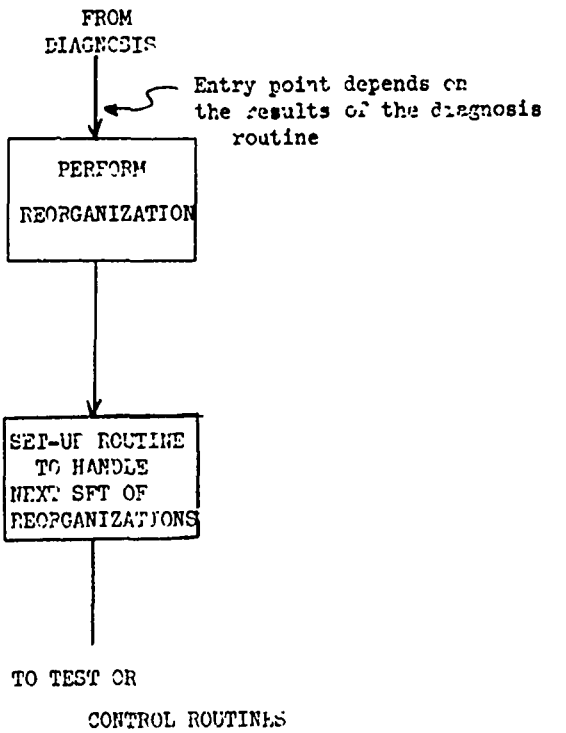
After reorganization is achieved, the self-reorganization routine checks the state of the state register and updates itself.

From the above description, it is seen that the master may consist of a counter which is used to sequence itself thru a short wired program and some one-shots, as shown in Figure 3.



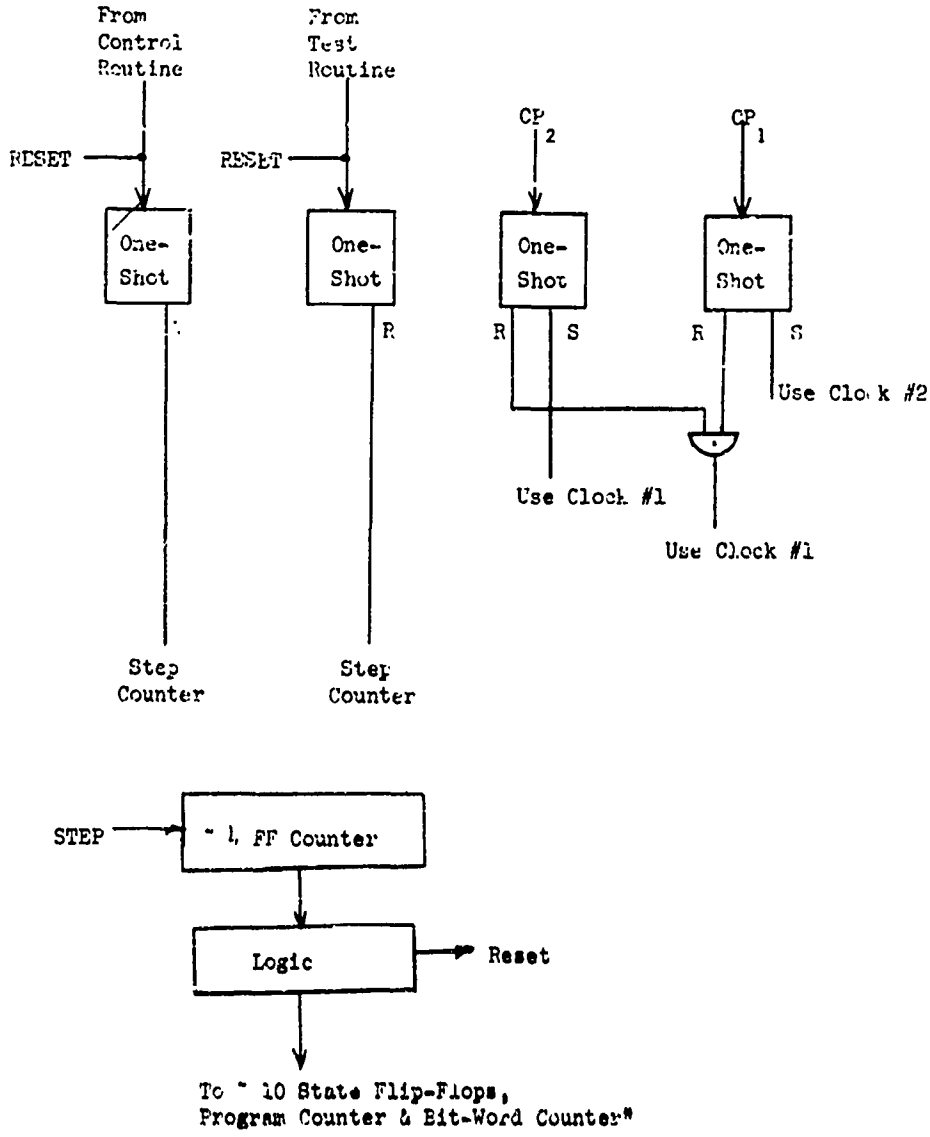
OVER-ALL PROGRAM FLOW CHART

FIGURE 1



SELF-REORGANIZATION ROUTINE

FIGURE 2



BLOCK DIAGRAM OF
MASTER UNIT

FIGURE 3

*The PC and B-WC are always restarted at the same States.